UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/690,056	10/20/2003	Darryl J. Gove	SUNMP357	3504

32291 7590 01/18/2007
MARTINE PENILLA & GENCARELLA, LLP
710 LAKEWAY DRIVE
SUITE 200
SUNNYVALE, CA 94085

EXAMINER	
PHAM, THAI V	
ART UNIT	**PAPER NUMBER**
2192	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	01/18/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

Office Action Summary	Application No.	Applicant(s)	
	10/690,056	GOVE, DARRYL J.	
	Examiner	Art Unit	
	Thai Van Pham	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 October 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-26* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-26* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *20 October 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

This is the initial office action based on the application filed on 10/20/2003.

Priority date that has been considered for this application is 10/20/2003.

Claims 1 – 26 are currently pending and have been considered below.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

1. Claims 1, 2, 8 – 10, 12, 13, 15 – 19, 21, 22, and 24 – 26 are rejected under 35 U.S.C. 102(e) as being anticipated by **Madsen et al.** (US 7,137,105).

-- Claim 1.

Madsen discloses *a method for obtaining traces of a program, comprising*:

- *executing an original set of instructions*;

(Figs. 2 and 14, item 28 – "main program" – and associated text, e.g., Col. 3: lines 38 – 41; "...monitoring software code as it is executed by a target CPU... ".)

- *switching execution from the original set of instructions to an instrumented version of the original set of instructions*; and

(Fig. 14 and associated text, e.g., Col.11: lines 29 – 33; "...instrumentation block **9'** may operate by using replacement module **24'** to replace the first instruction ...of program code **28** with other types of instructions that cause a change in program flow... ".)

• *generating traces through execution of one or more instrumentation instructions contained within the instrumented version of the original set of instructions.*

(Fig. 14, item 97 – "output to buffer" – and associated text, e.g., Col. 12: lines 1 – 8; "...routine **35'** may include one or more output instructions **97**, such as a "printf" statement, that generate output to a trace buffer **103**... ".)

-- Claim 2.

Madsen discloses *the method for obtaining traces of a program as recited in claim 1, wherein*

• *the switching of execution from the original set of instructions to the instrumented version of the original set of instructions occurs at a location of known state in the original set of instructions.*

(Fig. 14 and associated text, e.g., Col. 11: lines 29 – 60; "...instructions that cause a change in program flow (such as a "branch" or "jump to subroutine" instruction **30'**) ...Translation table **150** includes the original instruction along with additional information intended to facilitate restoring the program code **28** to its original state once the instrumentation is removed (e.g., after monitoring has been completed) ... ". Here, the points where the switching of execution from the original program to instrumented code as well as the return of execution from the instrumented code back to the original program occur at known states in the original program.)

-- Claim 8.

Madsen discloses *the method for obtaining traces of a program as recited in claim 1, wherein*

• *execution of the instrumented version of the original set of instructions is performed by an emulator.*

(Table 2 and associated text, e.g., Col. 6: lines 47 – 51; "...the processor executes steps 3 and 4 on the external bus where they are visible by the conventional bus/state analyzer, i.e., an emulator/debugging hardware or software system.")

-- Claim 9.

Madsen discloses *a method for obtaining traces of a program, comprising*:

• *executing an original code*;

(Figs. 2 and 14, item 28 – "main program" – and associated text, e.g., Col. 3: lines 38 – 41; "...monitoring software code as it is executed by a target CPU... ".)

• *switching execution from the original code to an instrumented code*;

(Fig. 14 and associated text, e.g., Col.11: lines 29 – 33; "...instrumentation block **9'** may operate by using replacement module **24'** to replace the first instruction ...of program code **28** with other types of instructions that cause a change in program flow... ".)

• *executing the instrumented code*;

(Fig. 2, Table 2: steps 5a – 5f, and associated text, e.g., Col. 7: lines 9 – 21; "Having decoded the misalignment instructions, steps 5a - 5f will execute the original replaced instructions and take any additional appropriate action ...".)

• *generating traces*; and

(Fig. 14, item 97 – "output to buffer" – and associated text, e.g., Col. 12: lines 1 – 8; "...routine **35'** may include one or more output instructions **97**, such as a "printf" statement, that generate output to a trace buffer **103**... ".)

• *switching execution from the instrumented code to the original code.*

(Fig. 14 and associated text, e.g., Col. 11: lines 29 – 60; "...instructions that cause a change in program flow (such as a "branch" or "jump to subroutine" instruction **30'**) ...Translation table **150** includes the original instruction along with additional information intended to facilitate restoring the program code **28** to its original state once the instrumentation is removed (e.g., after monitoring has been completed) ... ".)

-- Claim 10.

Madsen discloses *the method for obtaining traces of a program as recited in claim 9, further comprising*:

• *triggering the switching of execution from the original code to the instrumented code, the triggering causing the switching of execution to occur at a next location of known state in the original code.*

(Fig. 14 and associated text, e.g., Col. 11: lines 29 – 60; "...instructions that cause a change in program flow (such as a "branch" or "jump to subroutine" instruction **30'**) ...Translation table **150** includes the original instruction along with additional information intended to facilitate restoring the program code **28** to its original state once the instrumentation is removed (e.g., after monitoring has been completed) ... ". Here, the points where the switching of execution from the original program to instrumented code

as well as the return of execution from the instrumented code back to the original program occur at known states in the original program.)

-- Claim 12.

Madsen discloses *the method for obtaining traces of a program as recited in claim 9, further comprising*:

• *triggering the switching of execution from the instrumented code to the original code, the triggering causing the switching of execution to occur at a next location of known state in the instrumented code.*

(Fig. 14 and associated text, e.g., Col. 11: lines 29 – 60; "...instructions that cause a change in program flow (such as a "branch" or "jump to subroutine" instruction **30'**) ...Translation table **150** includes the original instruction along with additional information intended to facilitate restoring the program code **28** to its original state once the instrumentation is removed (e.g., after monitoring has been completed) ... ". Here, the points where the switching of execution from the original program to instrumented code as well as the return of execution from the instrumented code back to the original program occur at known states in the original program.)

-- Claim 13.

Madsen discloses *the method for obtaining traces of a program as recited in claim 12, wherein*

• *the next location of known state in the instrumented code corresponds to an instruction common to both the instrumented code and the original code.*

(Fig. 14 and associated text, e.g., Col. 11: lines 29 – 45; "...Replacement module **24'** may then insert the original (e.g. first) instruction into a translation table (also referred to as an instrumentation table) **150**... ".)

-- Claim 15.

Madsen discloses *the method for obtaining traces of a program as recited in claim 9, wherein*

• *both switching execution from the original code to the instrumented code and switching execution from the instrumented code to the original code are performed using return addresses during processing of function calls.*

(Fig. 9 – "instrument and Trace Function(s) with Calls" – and Fig. 11 – "instrument and Trace Function(s) without Calls" – and associated text.

Fig. 14, table 150 and associated text, e.g., Col. 11: lines 29 – 60; "...Translation table **150** includes the original instruction along with additional information intended to facilitate restoring the program code **28** to its original state once the instrumentation is removed (e.g., after monitoring has been completed) ... ". Using the address of the original opcode and address of scratch pad recorded in translation table, "branch" or "jump to subroutine" from the original program to instrumented code and return from the instrumented code back to the original program are properly executed.)

-- Claim 16.

Madsen discloses *the method for obtaining traces of a program as recited in claim 9, further comprising*:

• *defining a map of instruction addresses, the map of instruction addresses identifying correspondences between instruction addresses in the original code and instruction addresses in the instrumented code.*

(Fig. 14, table 150 and associated text, e.g., Col. 11: lines 29 – 60; "...Translation table **150** includes the original instruction along with additional information intended to facilitate restoring the program code **28** to its original state once the instrumentation is removed (e.g., after monitoring has been completed) ... ".)

-- Claim 17.

Madsen discloses *the method for obtaining traces of a program as recited in claim 16, wherein*

• *both switching execution from the original code to the instrumented code and switching execution from the instrumented code to the original code are performed using the map of instruction addresses.*

(Fig. 7 – "Set Breakpoint/Tracepoint at Entry Address" and "Set Breakpoint/Tracepoint in Address Range" – and associated text.

Fig. 14, table 150 and associated text, e.g., Col. 11: lines 29 – 60; "...Translation table **150** includes the original instruction along with additional information intended to facilitate restoring the program code **28** to its original state once the instrumentation is removed (e.g., after monitoring has been completed) ... ". Using the address of the original opcode and address of scratch pad recorded in translation table, "branch" or "jump to subroutine" from the original program to instrumented code and return from the instrumented code back to the original program are properly executed.)

-- Claims 18, 19, 21, 22, and 24 – 26.

Madsen discloses a computer product (Col. 2: lines 21 – 35; "computer usable medium having computer readable program code embodied therein...") for performing a method corresponding to the method of claims 9, 10, 12, 13, and 15 – 17, respectively;

Therefore, claims 18, 19, 21, 22, and 24 – 26 are rejected for the same reason set forth in connection to the rejection of claims 9, 10, 12, 13, and 15 – 17 above, respectively.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 3 – 7, 11, 14, 20, and 23 are rejected under 35 U.S.C. 103(a) as being obvious over **Madsen et al.** (US 7,137,105).

-- Claim 3.

Madsen discloses *the method for obtaining traces of a program as recited in claim 1*, however, does not disclose the method further comprising:

• *triggering the switching of execution from the original set of instructions to the instrumented version of the original set of instructions, wherein the triggering is based on an elapsed time of execution, the triggering causing the switching of execution to occur at a next location of known state in the original set of instructions.*

Madsen discloses that the "instruction locating module" is capable of locating the designated instructions for instrumentation within a specified address range(s) (Fig.1 and associated text, e.g., Col. 4: lines 24 – 26; "Instruction locating module 22 scans a block of program code within a user designated address range or ranges to locate predetermined instructions."). Furthermore, **Madsen** discloses that execution time of every function and/or instruction can be recorded (Figs. 10, and 12 and associated text, e.g. Col. 9: lines 29 – 61). In the function "dayofYear", nested functions and the function itself are instrumented and traced with corresponding elapsed execution time recorded. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to set instrumentation branches in the main program, i.e., triggering points for switching execution from the original set of instructions to the instrumented version of the original set of instructions, based on an elapsed time of execution in order to probe the program execution for examination of its behavior at a particular time interval of execution as traced and recorded in **Madsen** (e.g., "dayofYear" function in Figs. 10.and 12).

-- Claim 4.

Madsen discloses *the method for obtaining traces of a program as recited in claim 3*, however, does not further disclose

• *the triggering is performed such that execution of the original set of instructions accounts for more than about 90 percent of the elapsed time of execution.*

Madsen discloses that execution time of every function and/or instruction can be recorded (Figs. 8, 10, and 12 and associated text). As a result, the total execution time

of the entire program or of any particular portion of the program, e.g., a program function or block, would be easily and simply calculated. The elapsed time of execution is a subset of the different types of execution time **Madsen**'s invention is capable of keeping track of.

Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to set the elapsed time of execution to any percentage of execution time, e.g., 90%, in order to probe the program execution for examination of its behavior after a predetermined amount of time of normal execution has completed.

-- Claim 5.

Madsen discloses *the method for obtaining traces of a program as recited in claim 1*, further comprising

• *switching execution from the instrumented version of the original set of instructions back to the original set of instructions.*

(Fig. 14 and associated text, e.g., Col. 11: lines 29 – 60; "...instructions that cause a change in program flow (such as a "branch" or "jump to subroutine" instruction **30'**) ...Translation table **150** includes the original instruction along with additional information intended to facilitate restoring the program code **28** to its original state once the instrumentation is removed (e.g., after monitoring has been completed) ... ".)

Madsen, however, does not disclose the method further comprising:

• *triggering a switching of execution from the instrumented version of the original set of instructions back to the original set of instructions, wherein the triggering is based on an elapsed time of execution, the triggering causing the switching of execution to occur at a*

next location of known state in the instrumented version of the original set of instructions.

Madsen discloses that the "instruction locating module" is capable of locating the designated instructions for instrumentation within a specified address range(s) (Fig.1 and associated text, e.g., Col. 4: lines 24 – 26; "Instruction locating module 22 scans a block of program code within a user designated address range or ranges to locate predetermined instructions."). Furthermore, **Madsen** discloses that execution time of every function and/or instruction can be recorded (Figs. 8, 10, and 12 and associated text, e.g. Col. 9: lines 29 – 61). In the function "dayofYear", the Entry/Exit of the function, nested functions and the function itself are instrumented and traced with corresponding elapsed execution time recorded.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to set instrumentation branches in the main program, i.e., triggering points for switching execution from the original set of instructions to the instrumented version of the original set of instructions, based on an elapsed time of execution in order to probe the program execution for examination of its behavior at a particular time interval of execution as traced and recorded in **Madsen** (e.g., "dayofYear" function in Figs. 8, 10.and 12).

-- Claim 6.

Madsen discloses *the method for obtaining traces of a program as recited in claim 5, wherein*

• *the next location of known state in the instrumented version of the original set of instructions corresponds to an instruction common to both the instrumented version of the original set of instructions and the original set of instructions.*

(Fig. 14 and associated text, e.g., Col. 11: lines 29 – 45; "...Replacement module **24'** may then insert the original (e.g. first) instruction into a translation table (also referred to as an instrumentation table) **150**... ".)

-- Claim 7.

Madsen discloses *the method for obtaining traces of a program as recited in claim 5*, however, does not further disclose

• *the triggering is performed such that execution of the instrumented version of the original set of instructions accounts for less than about 10 percent of the elapsed time of execution.*

Madsen discloses that execution time of every instrumented function and/or instruction can be recorded (Figs. 8, 10, and 12 and associated text). As a result, the total execution time of the entire program or of any particular instrumented portion of the program, e.g., an instrumented program function or block, would be easily and simply calculated. The elapsed time of execution is a subset of the different types of execution time **Madsen**'s invention is capable of keeping track of.

Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to set the elapsed time of execution to any percentage of execution time, e.g., 10%, in order to probe the program execution for examination of its behavior during a predetermined amount of time of normal execution.

-- Claim 11.

Madsen discloses *the method for obtaining traces of a program as recited in claim I0*, however, does not further disclose

• *the triggering is based on an elapsed time of execution*;

Madsen discloses that the "instruction locating module" is capable of locating the designated instructions for instrumentation within a specified address range(s) (Fig.1 and associated text, e.g., Col. 4: lines 24 – 26; "Instruction locating module 22 scans a block of program code within a user designated address range or ranges to locate predetermined instructions."). Furthermore, **Madsen** discloses that execution time of every function and/or instruction can be recorded (Figs. 8, 10, and 12 and associated text, e.g. Col. 9: lines 29 – 61). In the function "dayofYear", the Entry/Exit of the function, nested functions and the function itself are instrumented and traced with corresponding elapsed execution time recorded.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to set instrumentation branches in the main program, i.e., triggering points for switching execution from the original set of instructions to the instrumented version of the original set of instructions, based on an elapsed time of execution in order to probe the program execution for examination of its behavior at a particular time interval of execution as traced and recorded in **Madsen** (e.g., "dayofYear" function in Figs. 10.and 12).

• *the triggering being performed such that execution of the original code accounts for more than about 90 percent of the elapsed time of execution.*

Madsen discloses that execution time of every function and/or instruction can be recorded (Figs. 8, 10, and 12 and associated text). As a result, the total execution time of the entire program or of any particular portion of the program, e.g., a program function or block, would be easily and simply calculated. The elapsed time of execution is a subset of the different types of execution time **Madsen**'s invention is capable of keeping track of.

Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to set the elapsed time of execution to any percentage of execution time, e.g., 90%, in order to probe the program execution for examination of its behavior after a predetermined amount of time of normal execution has completed.

-- Claim 14.

Madsen discloses *the method for obtaining traces of a program as recited in claim 12*, however, does not further disclose

- *the triggering is based on an elapsed time of execution*;

Madsen discloses that the "instruction locating module" is capable of locating the designated instructions for instrumentation within a specified address range(s) (Fig.1 and associated text, e.g., Col. 4: lines 24 – 26; "Instruction locating module 22 scans a block of program code within a user designated address range or ranges to locate predetermined instructions."). Furthermore, **Madsen** discloses that execution time of every function and/or instruction can be recorded (Figs. 8, 10, and 12 and associated text, e.g. Col. 9: lines 29 – 61). In the function "dayofYear", the Entry/Exit of the

function, nested functions and the function itself are instrumented and traced with corresponding elapsed execution time recorded.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to set instrumentation branches in the main program, i.e., particularly return to main program from instrumented code or triggering points for switching execution from the instrumented version of the original set of instructions back to the original set of instructions, based on an elapsed time of execution in order to probe the program execution for examination of its behavior at a particular time interval of execution.

• *the triggering being performed such that execution of the instrumented code accounts for less than about 10 percent of the elapsed time of execution.*

Madsen discloses that execution time of every instrumented function and/or instruction can be recorded (Figs. 8, 10, and 12 and associated text). As a result, the total execution time of the entire program or of any particular instrumented portion of the program, e.g., an instrumented program function or block, would be easily and simply calculated. The elapsed time of execution is a subset of the different types of execution time **Madsen**'s invention is capable of keeping track of.

Thus, it would have been obvious to one of ordinary skill in the art at the time the invention was made to set the elapsed time of execution to any percentage of execution time, e.g., 10%, in order to probe the program execution for examination of its behavior during a predetermined amount of time of normal execution.

-- Claims 20 and 23.

Madsen discloses a computer product (Col. 2: lines 21 – 35; "computer usable medium having computer readable program code embodied therein...") of claim 19 and 20, respectively, for performing a method corresponding to the method of claims 11 and 14, respectively; Therefore, claims 20 and 23 are rejected for the same reason set forth in connection to the rejection of claims 11 and 14 above, respectively.

Conclusion

The prior art made of record and not relied upon is considered pertinent to Applicant's disclosure. See the attached Notice of References Cited.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Thai Van Pham whose telephone number is (571) 270-1064. The examiner can normally be reached on Monday - Thursday, 8am - 3pm EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

TVP

TUAN DAM
SUPERVISORY PATENT EXAMINER